

المحاضرة الثانية د / مجدي الشرقاوي

#include "stdafx.h" الأضافة

#include <iostream>

using namespace std;

//----- class IntNode for creating new node -----//

class IntNode {

public :

IntNode(int el, IntNode *ptr = 0) {info = el; next = ptr;}

int info;

IntNode *next;

};

//----- class IntLList for dealing with nodes -----//

class IntLList {

public:

IntLList() {head = tail = 0; }

void AddToHead(int);

void AddToTail(int);

void DisplayList();

private:

IntNode *head, *tail;

};

void IntLList::AddToHead(int data)

{

IntNode *newnode;

newnode = new IntNode(data, 0);

newnode->next = head;

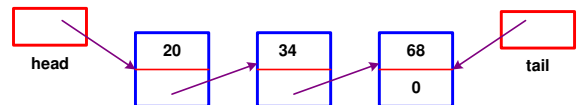
head = newnode;

if (tail==0)

tail = head;

}

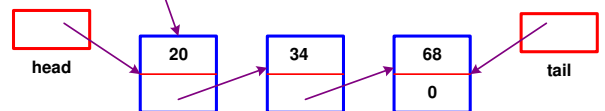
أضافة عنصر الى رأس القائمة
أضافة عنصر الى ذيل القائمة
عرض عناصر القائمة



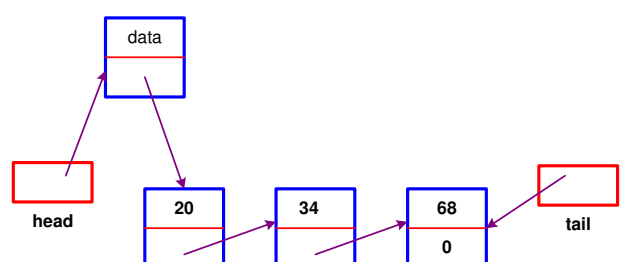
new node



new node



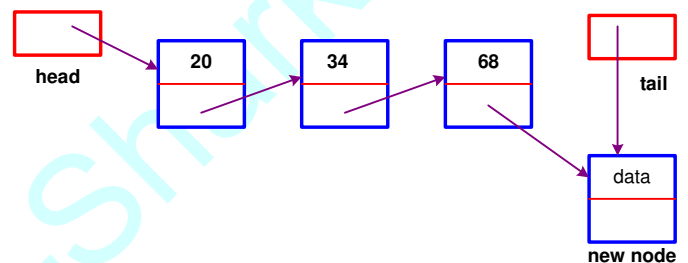
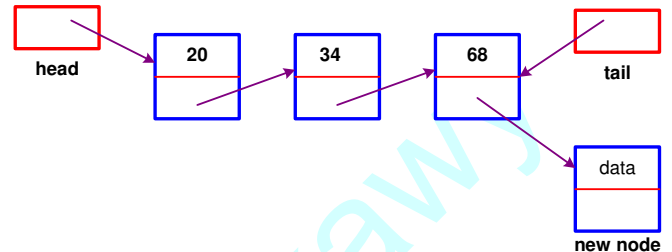
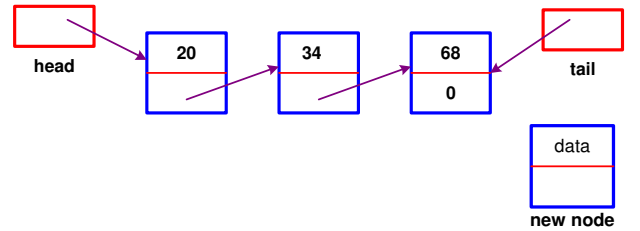
new node



```

void IntLList::AddToTail(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data, 0);
    tail->next = newnode;
    tail = newnode;
    if (head==0)
        head = tail;
}

```



```

void IntLList::DisplayList()
{
    IntNode *current;
    current = head;
    cout << "head = " << head << "\n";
    while(current != 0)
    {
        cout << current->info << " " << current << "\n";
        current=current->next;
    }
    cout << "tail = " << tail << "\n";
    cout << "-----" << "\n";
}

```

```

void main()
{
    IntLList mag;
    mag.AddToHead(50);
    mag.AddToHead(90);
    mag.AddToHead(60);
    mag.DisplayList();
    mag.AddToTail(88);
    mag.AddToTail(77);
    mag.AddToHead(66);
    mag.DisplayList();
}

```

```

C:\Windows\system32\cmd.exe
head = 00061A00
60 00061A00
90 000619C8
50 00061990
tail = 00061990
-----
head = 00241FC0
66 00241FC0
60 00061A00
90 000619C8
50 00061990
88 00241F50
77 00241F88
tail = 00241F88
-----
Press any key to continue . . .

```

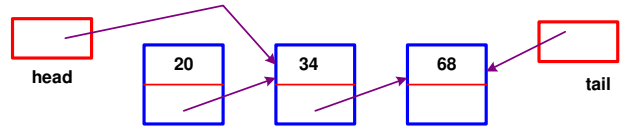
الألف

```
#include "stdafx.h"
#include <iostream>
using namespace std;
//----- class IntNode for creating new node -----//
class IntNode {
public:
    IntNode(int el, IntNode *ptr = 0) {info = el; next = ptr;}
    int info;
    IntNode *next;
};
//----- class IntLList for dealing with nodes -----//
class IntLList {
public:
    IntLList() {head = tail =0; }
    void AddToHead(int);
    void AddToTail(int);
    void DeleteFromHead();
    void DeleteFromTail();
    void DeleteNode(int);
    void DisplayList();
private:
    IntNode *head, *tail;
};
void IntLList::AddToHead(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    newnode->next = head;
    head = newnode;
    if (tail==0)
        tail = head;
}
void IntLList::AddToTail(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    tail->next = newnode;
    tail = newnode;
    if (head==0)
        head = tail;
}
```

```

void IntLList::DeleteFromHead()
{
    if(head!=0)
        head = head->next;
}

```

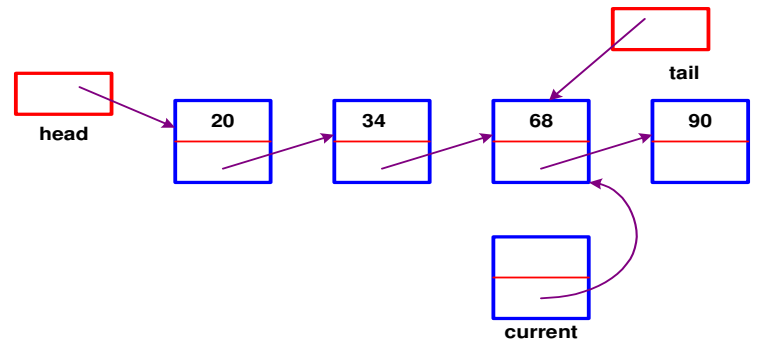
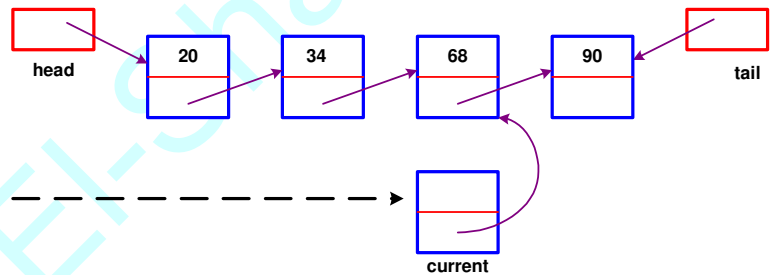
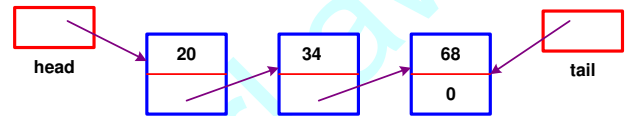


```

void IntLList::DeleteFromTail()
{
    IntNode *current;
    current = head;
    while(current->next != tail)
    {
        current=current->next;
    }
    tail=current;
}

```

ننشأ عقده جديده ونجعلها تشير الى راس السلسله ثم نحركها حتى تشير الى الحلقة القبل الأخيره ثم نجعل الذيل يشير الى الحلقة القبل الأخيره



```

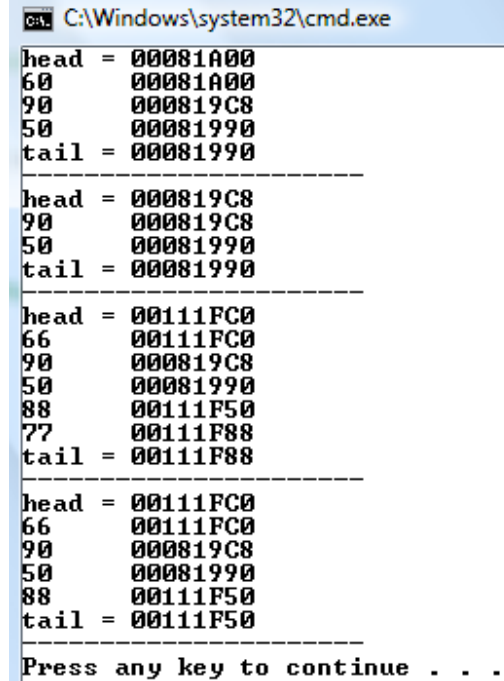
void IntLList::DisplayList()
{
    IntNode *current;
    current = head;
    cout << "head = " << head << "\n";
    while(current != 0)
    {
        cout << current->info << " " << current << "\n";
        current=current->next;
    }
    cout << "tail = " << tail << "\n";
    cout << "-----" << "\n";
}

```

```

void main()
{
    IntLList mag;
    mag.AddToHead(50);
    mag.AddToHead(90);
    mag.AddToHead(60);
    mag.DisplayList();
    mag.DeleteFromHead();
    mag.DisplayList();
    mag.AddToTail(88);
    mag.AddToTail(77);
    mag.AddToHead(66);
    mag.DisplayList();
    mag.DeleteFromTail();
    mag.DisplayList();
}

```



```

C:\Windows\system32\cmd.exe
head = 00081A00
60 00081A00
90 000819C8
50 00081990
tail = 00081990
-----
head = 000819C8
90 000819C8
50 00081990
tail = 00081990
-----
head = 00111FC0
66 00111FC0
90 000819C8
50 00081990
88 00111F50
77 00111F88
tail = 00111F88
-----
head = 00111FC0
66 00111FC0
90 000819C8
50 00081990
88 00111F50
tail = 00111F50
-----
Press any key to continue . . .

```

البحث عن قيمة معينة داخل السلسلة `#include "stdafx.h"`

```
#include <iostream>
using namespace std;
//----- class IntNode for creating new node -----//
class IntNode {
public :
    IntNode(int el, IntNode *ptr = 0) {info = el; next = ptr;}
    int info;
    IntNode *next;
};
//----- class IntLList for dealing with nodes -----//
class IntLList {
public:
    IntLList() {head = tail =0; }
    void AddToHead(int);
    void AddToTail(int);
    void DeleteFromHead();
    void DeleteFromTail();
    void DeleteNode(int);
    void FindNode(int);
    void DisplayList();
private:
    IntNode *head, *tail;
};

void IntLList::AddToHead(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    newnode->next = head;
    head = newnode;
    if (tail==0)
        tail = head;
}

void IntLList::AddToTail(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    tail->next = newnode;
    tail = newnode;
}
```

```

void IntLList::DeleteFromHead()
{
    if(head!=0)
        head = head->next;
    if(head==0)
        tail=head;
}

void IntLList::DeleteFromTail()
{
    IntNode *current;
    current = head;
    while(current->next != tail)
    {
        current=current->next;
    }
    tail=current;
    tail->next=0;
}

void IntLList::FindNode(int val)
{
    IntNode *current;
    current = head;
    while(current != tail->next)
    {
        if (current->info == val)
            cout << current->info << " " << current << "\n";
        current=current->next;
    }
}

void IntLList::DisplayList()
{
    IntNode *current;
    current = head;
    cout << "head = " << head << "\n";
    while(current != tail->next)
    {
        cout << current->info << " " << current << "\n";
        current=current->next;
    }
    cout << "tail = " << tail << " " << tail->next << "\n";
    cout << "-----" << "\n";
}

```

```

void main()
{
    IntLList mag;
    mag.AddToHead(50);
    mag.AddToHead(90);
    mag.AddToHead(60);
    mag.DisplayList();
    mag.DeleteFromHead();
    mag.DisplayList();
    mag.AddToTail(88);
    mag.AddToTail(77);
    mag.AddToHead(66);
    mag.AddToHead(50);
    mag.DisplayList();
    mag.DeleteFromTail();
    mag.DisplayList();
    mag.FindNode(50);
}

```

```

C:\Windows\system32\cmd.exe
head = 008B1A00
60    008B1A00
90    008B19C8
50    008B1990
tail = 008B1990 00000000
-----
head = 008B19C8
90    008B19C8
50    008B1990
tail = 008B1990 00000000
-----
head = 00181D40
50    00181D40
66    00181FC0
90    008B19C8
50    008B1990
88    00181F50
77    00181F88
tail = 00181F88 00000000
-----
head = 00181D40
50    00181D40
66    00181FC0
90    008B19C8
50    008B1990
88    00181F50
tail = 00181F50 00000000
-----
50    00181D40
50    008B1990
Press any key to continue . . .

```

Dr. Magdi El

البحث عن قيمة معينة والغائها

```
#include "stdafx.h"
#include <iostream>
using namespace std;
//----- class IntNode for creating new node -----//
class IntNode {
public :
    IntNode(int el, IntNode *ptr = 0) {info = el; next = ptr;}
    int info;
    IntNode *next;
};
//----- class IntLList for dealing with nodes -----//
class IntLList {
public:
    IntLList() {head = tail =0; }
    void AddToHead(int);
    void AddToTail(int);
    void DeleteFromHead();
    void DeleteFromTail();
    void DeleteNode(int);
    void FindNode(int);
    void DelNode(int);
    void DisplayList();
private:
    IntNode *head, *tail;
};

void IntLList::AddToHead(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    newnode->next = head;
    head = newnode;
    if (tail==0)
        tail = head;
}

void IntLList::AddToTail(int data)
{
    IntNode *newnode;
    newnode = new IntNode(data,0);
    tail->next = newnode;
    tail = newnode;
}
```

```
void IntLList::DeleteFromHead()
{
    if(head!=0)
        head = head->next;
    if(head==0)
        tail=head;
}
```

```
void IntLList::DeleteFromTail()
{
    IntNode *current;
    current = head;
    while(current->next != tail)
    {
        current=current->next;
    }
    tail=current;
    tail->next=0;
}
```

```
void IntLList::FindNode(int val)
{
    IntNode *current;
    current = head;
    while(current != tail->next)
    {
        if (current->info == val)
            cout << current->info << " " << current << "\n";
        current=current->next;
    }
    cout << "-----" << "\n";
}
```

```

void IntLList::DelNode(int val)
{
    IntNode *current, *previous;
    current = previous = head;
    if(head->info==val)
        head = current->next;
    while(current != tail->next)
    {
        if (current->info == val)
        {
            previous->next = current->next;
        }
        previous = current;
        current=current->next;
    }
}

void IntLList::DisplayList()
{
    IntNode *current;
    current = head;
    cout << "head = " << head << "\n";
    while(current != tail->next)
    {
        cout << current->info << " " << current << "\n";
        current=current->next;
    }
    cout << "tail = " << tail << " " << tail->next << "\n";
    cout << "-----" << "\n";
}

```

```

void main()
{
    IntLList mag;
    mag.AddToHead(50);
    mag.AddToHead(90);
    mag.AddToHead(60);
    mag.DisplayList();
    mag.DeleteFromHead();
    mag.DisplayList();
    mag.AddToTail(88);
    mag.AddToTail(77);
    mag.AddToHead(66);
    mag.AddToHead(50);
    mag.DisplayList();
    mag.DeleteFromTail();
    mag.DisplayList();
    mag.FindNode(50);
    mag.DelNode(50);
    mag.DisplayList();
}

```

```

C:\Windows\system32\cmd.exe

head = 00801A00
60    00801A00
90    008019C8
50    00801990
tail = 00801990  00000000
-----
head = 008019C8
90    008019C8
50    00801990
tail = 00801990  00000000
-----
head = 006A1D40
50    006A1D40
66    006A1FC0
90    008019C8
50    00801990
88    006A1F50
77    006A1F88
tail = 006A1F88  00000000
-----
head = 006A1D40
50    006A1D40
66    006A1FC0
90    008019C8
50    00801990
88    006A1F50
tail = 006A1F50  00000000
-----
50    006A1D40
50    00801990
-----
head = 006A1FC0
66    006A1FC0
90    008019C8
88    006A1F50
tail = 006A1F50  00000000
-----

```